Program Test Plan

Deliverable ID:	*I-MANAGE Program Test Plan*
Version number:	*1.0*
Draft/Final as of:	*16 January 2004*
Printed on:	*16 January 2004*
Author:	*Rob Briede*
Owner:	*Rob Briede*

Status of Document

☐ Draft

☒ Delivered

☒ Accepted

Document information

Document source

This document is maintained as an online document. Contact the author for the latest version.

Revision history

Version number	Date	Summary of changes	Revised By

Approvals

The following people have approved this document. (Sign below name)

Name	Function
Warren Huffer	I-MANAGE Program Federal Executive
Signature	**Date:** 1/20/2004
Christopher Simpson	I-MANAGE Program Manager
Signature:	**Date:** 1/20/2004

Distribution

This document has been distributed to:

Name	Function

Table of Contents

1. Purpose

The I-MANAGE Program Test Plan prescribes the basic steps and testing to be accomplished to have a viable test program. It describes general guidance for each project within the I-MANAGE portfolio. Each project within I-MANAGE may vary by software type, acquisition strategy, stage of maturity, management method and other characteristics. Each project will have a specific test plan that is valid for only that project. Project Managers are responsible for the creation and exercise of the test plans for their project.

1.1 Program Objectives

The Department launched the Integrated Management Navigation System, I-MANAGE, in January 2003, as a cost-saving program designed to use technology to meet the challenge for achieving greater management efficiencies. The I-MANAGE Program is a key cornerstone in the Department's efforts to implement improved financial performance, integrated budget and

performance, and expanded electronic government, as outlined in the President's Management Agenda.

1.2 System Description

The I-MANAGE Program will consolidate and streamline the Department's business systems by integrating management information related to financial and cost accounting, travel, payroll, budget formulation and execution, procurement and contracts management, facilities management, human resources, and research and development. Initial efforts will focus on various system components to include the Standard Accounting and Reporting System (STARS), the I-MANAGE Data Warehouse, the Standard Budget System, E-Procurement, and the Human Resource Information System (CHRIS). IBM is the Prime Contractor for the STARS Project.

1.3 Plan Objectives

1.3.1 To ensure a consistent and auditable approach to early and continual software process verification

1.3.2 To ensure that an effective approach to formal testing is developed that exercises the components of I-MANAGE systems and the functional and technical aspects of the associated system's interfaces.

1.3.3 To assist the I-MANAGE Project Managers in making decisions regarding the assignment and deployment of human and physical resources, roles and responsibilities, and the test management procedures required in support of testing.

1.3.4 To support project quality assurance processes.

1.3.5 To develop an overarching test schedule that recognizes timeframes and milestones for testing by each and all I-MANAGE projects.

1.4 References

1. DOE's Software Engineering Methodology (SEM),
2. McCabe, Thomas J. & Butler, Charles W. "Design Complexity Measurement and Testing." December 1989
3. I-MANAGE Software Quality Assurance Plan, January 2004
4. I-MANAGE STARS Cross Accountability Matrix, December 2003
5. I-MANAGE IDW Cross Accountability Matrix, December 2003

1.5 Outstanding Issues

None as of publication of this version of the document.

2. Test Scope

2.1 Features to be Tested

The scope of testing within I-MANAGE will take different forms for the various projects that are an on-going effort throughout the Department of Energy. In most cases, due to the nature of COTS software packages, design and production testing may the responsibility of the prime contractor or subcontractors with minimal testing being required by DOE project personnel. Features to be tested are broken into the following categories:

2.1.1 System Technical and Functional Accuracy: Testing that the system is performing as specified and designed

2.1.2 Numerical Accuracy: Testing that the system not only performs correctly, but is to the specified accounting numerical accuracy.

2.1.3 Platform Stability: Testing that the system operates under various loads and conditions.

2.2 Types of Tests

At a minimum, the following testing will be conducted:

2.2.1 Functional Application Unit Testing intended to verify the quality and integrity of the systems functionality from an end-user perspective. Unit tests will be conducted on individual COTS software modules or packages.

2.2.2 Functional Application Integration Testing to expand on the Unit Testing. This testing will be conducted to ensure that each of modules and applications work together as designed and required.

2.2.3 Human Interface and Usability Testing that ensures that the software supports inputs and outputs that provide usability to the targeted users.

2.2.4 Security Testing to ensure the system software, hardware and stored data and information are protected from internal and external threats.

2.2.5 Documentation Testing to verify that Users Manual and other formal documentation is accurate and will provide DOE with the capability to operate the system.

2.2.6 User Acceptance Testing to finalize all development and to place the system into production.

2.2.7 Regression Testing to ensure changes made to the software both 1. work correctly and 2. have not changed detrimentally, other features of the software.

2.3 Features Not to be Tested

None, as of publication of this version of the document.

3. Test Methodologies

3.1 Testing Approach

The testing approach for each system, module and project within I-MANAGE will be tailored to the acquisition strategy, (GOTS, COTS or custom built), the type of management used (internal, contracted, partnership), the complexity of the system and other factors. The testing tools will vary similarly as the approach and resources available for each project. Each Project Manager shall develop a Master Test Plan for their project that provides in detail the approach taken for that particular project.

In general and at a minimum the following testing shall be conducted:

3.1.1 Technical Unit Testing (Non-COTS code and interfaces)
- 3.1.1.1 Path Testing: Test each logic branch
- 3.1.1.2 Boundary Condition Testing: Test limits and parameter tolerances
- 3.1.1.3 Input Validation and Syntax Testing: Test error handling
- 3.1.1.4 State Transition Testing: Test to uncover incorrect or unreachable states
- 3.1.1.5 Database Testing: Testing of COTS data transition

3.1.2 Function Unit Testing (COTS modules)
- 3.1.2.1 Boundary Condition Testing: Test limits and parameter tolerances
- 3.1.2.2 Transaction Flow Testing: Test each integrated unit and component for valid paths

3.1.3 Integration Testing
- 3.1.3.1 Input Validation and Syntax Testing: Test error handling of integrated modules.
- 3.1.3.2 Transaction Flow Testing: Test each integrated unit and component for valid paths and verify functionality against requirements.
- 3.1.3.3 Database Driven Testing: Black box testing of combined modules and external interfaces to determine problems with COTS interfaces, data corruption and impossible states.
- 3.1.3.4 State Transition Testing: Test to uncover incorrect or unreachable states

3.1.4 System/Platform Testing
- 3.1.4.1 Volume Testing: Test to determine the continuous load at which the system fails
- 3.1.4.2 Performance Testing: Determine actual system performance
- 3.1.4.3 Configuration Testing: Test each hardware and software configuration to be supported
- 3.1.4.4 Compatibility Testing: Test systems claimed compatibility with other software
- 3.1.4.5 Load/Stress Testing: Testing to determine peak load conditions in which the system fails.
- 3.1.4.6 Security Testing: Testing to identify system security weaknesses
- 3.1.4.7 Reliability and Availability Testing: Test to determine system up-time at typical loads

3.1.4.8 Platform Stability Testing: Test system operation and shutdown behavior during environment changes

3.1.5 Acceptance Testing: Test that system performs as designed and contracted

3.1.6 Regression Testing: Test affects of the system to changes within a module

3.2 Test Data

Test data generation may vary between the projects within I-MANAGE and will be tailored to the acquisition strategy, (GOTS, COTS or custom built), the type of management used (internal, contracted, partnership), the complexity of the system and other factors. Because of the accuracy requirements of I-MANAGE and the need to meet the requirement specifications, at a minimum, the following test data shall be used:

Test Scripts that test one or more of the project technical and functional requirements. Each Test Script shall be designed to test as many of the requirements as possible in each test. Additionally, the scripts shall be designed to ensure all aspect discussed in Section 3.1 above such as path testing and boundary condition testing.

3.3 Test Documents

I-MANAGE testing documentation begins with each Project Test Plan produced by each project team. At a minimum, each Project Test Plan will include:

3.3.1 Test Entry and Exit Criteria that delineates the environment that the tests will be conducted, what state in which the system will be conducted and formal procedures for exiting the test to ensure that the test is valid.

3.3.2 A Software Quality Assurance Plan that explains how the test processes within the project will be validated through outside verification.

3.3.3 A Cross-Accountability Plan that traces each system requirement with a test that will validate that the requirement is fulfilled.

3.3.4 Test Scripts as explained in section 3.2 above.

3.3.5 A Project Testing Signature Sign-Off Log that shows accountability for the conduct and completion of all tests performed within a project.

3.4 Requirements Validation

Requirements Validation will be conducted through the use of the Cross-Accountability Plan. This plan will trace each system requirement with a test that will validate that the requirement is fulfilled. The Cross-Accountability Plan shall present test and the associated test script or scripts in a matrix format using Microsoft Excel, Dbase, Word or other suitable software.

3.4 Control Procedures

Each test conducted shall use the Test Plan Entry and Exit Plan. The test environment shall be noted and recorded prior to performing any test in order to validate the test. After a test script is run, the results of the test will be documented with the test scrip. A report of test results and deviations shall be given to the Test Manager, the I-MANAGE Testing and QA Manager, Configuration Manager and the Project Manager. Further details on specific control procedures shall be provided in each individual Project Test Plan.

4. Test Phases

The number and type of test phases provided in each project within I-MANAGE will differ by a particular project's acquisition strategy, size, organization, purpose and other reasons. In general, projects within I-MANAGE shall have the following minimum test phases: Unit Testing, Integration Testing, System/Platform Testing, Acceptance Testing and Regression Testing. Brief definitions for each phase are provided in section 3.1 above and described in greater detailed in each individual Project Test Plan

4.1 Participants

Persons or groups responsible for conducting each test phase shall be provided in each individual Project Test Plan and also the Project Testing Sign-Off Log.

4.2 Sources of Data

Sources of data will vary by project and shall be described in each individual Project Test Plan.

4.3 Entrance and Exit Criteria

Entrance and exit criteria will vary by project and shall be described in each individual Project Test Plan.

4.4 Requirements

Project Requirements will vary by project and shall be described in each individual Project Test Plan.

5. Test Environment

5.1 Hardware

The hardware requirements for the test environment throughout the lifecycle of I-MANAGE will vary from project to project. Specific hardware requirements for each project shall be detailed in each Project Test Plan.

5.2 Software

The software requirements for the test environment throughout the lifecycle of I-MANAGE will vary from project to project. Specific software requirements for each project shall be detailed in each Project Test Plan.

5.3 Location

The location requirements, including physical plant structure for the test environment throughout the lifecycle of I-MANAGE will vary from project to project. Specific location requirements for each project shall be detailed in each Project Test Plan.

5.4 Staffing and Training

The staffing and training requirements for the test environment throughout the lifecycle of I-MANAGE will vary from project to project. Specific staffing and training requirements for each project shall be detailed in each Project Test Plan.

6. Schedule

Specific testing activities and responsibilities will be presented in each individual Project Test Plan. The I-MANAGE SQA Plan will be integrated into each Project Test Plan. Scheduling of testing activities shall be done in a manner that completes as many tests as possible, as soon as possible. Testing activities shall not be idle, while a test can be completed unless it would be more efficient in the long run to wait. The risk of waiting is that unexpected problems may appear that require substantial work in resolving. Waiting to conduct testing activities delays the opportunity to resolve problems and may have a detrimental effect on the project schedule.

Ultimately, each individual Project Manager is responsible for the scheduling and completion of testing their software and systems. The Project Manager is responsible for ensuring that the testing teams are fully resourced, trained and active in their testing activities. The I-MANAGE Test Manager shall act in a Program Level capacity as oversight of the Testing Program and Environment. The I-MANAGE Test Manager shall make recommendations to the Project Manager concerning scheduling and resourcing issues that may affect the testing schedule.